REMARKS

In this amendment, Applicants amend claims 1, 2, 21, 28, 29, 33, 35 and 36. Claim 3 is canceled without prejudice.

Claims 1, 2, 4, 5, 8-37, 39 and 42-47 are currently pending. Claims 1, 21, 28 and 37 are independent claims. Applicants respectfully request reconsideration of the outstanding rejections and passage of the claims to allowance in view of the comments set forth below.

I. Summary of the Outstanding Rejections

Claims 1-3, 5, 21-23, 26-33, 36, 37, 39 and 44-47 stand rejected under 35 USC § 103(a) as being unpatentable over Denk et al (U.S. Patent Publication No.: US 2001/0025292-A1) (hereinafter "Denk") in view of the "Digital Filter Solution" website (http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002) (hereinafter "DFS").

Claims 4, 24, 25, 34, 42 and 43 stand rejected under 35 USC § 103(a) as being unpatentable over Denk in view DFS and in further view of Gay (U.S. Patent No. 5,677,951).

Claims 8-13 and 15-20 stand rejected under 35 USC § 103(a) as being unpatentable over Denk in view DFS and in further view of the "AutoCode Solutions" website (http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/ , dated 2002) (hereinafter "AutoCode Solutions").

Claim 14 stands rejected under 35 USC § 103(a) as being unpatentable over Denk in view DFS, and AutoCode Solutions in further view of Velazquez (U.S. Patent Publication No.: 2002/0121993).

Claim 35 stands rejected under 35 USC § 103(a) as being unpatentable over Denk in view DFS in further view of Pickerd (U.S. Patent Publication No.: 2002/0147554).

II. Rejection of Claims 1-3, 5, 21-23, 26-33, 36, 37, 39, and 44-47 under 35 §USC 103(a).

Claims 1-3, 5, 21-23, 26-33, 36, 37, 39, and 44-47 stand rejected under 35 USC § 103(a) as being unpatentable over Denk in view of DFS. The Applicants respectfully traverse this rejection.

A. Claim 3

As claim 3 has been canceled without prejudice herein, the rejection of claim 3 is now moot.

B. Claims 1, 2 and 5

Amended independent claim 1 reads as follows:

1. In a computer system, a method for implementing and using a filter object, the method comprising:
providing the filter object in a dynamically typed text-based programming environment, the filter object including a state, and the filter object being represented by:
an output equation for generating an output of the filter object, and
a state equation for updating the state of the filter object;
implementing the filter object; and
determining an output of the filter object based on an input to the filter object in a first dynamically typed text-based programming environment, determining the output of the filter object including:
receiving the input at the filter object,
identifying a first state of the filter object,
splitting up the input into a first input and a second input,
invoking a first call to a filtering method of the filter object to:
process the output equation to determine a first output of the filter object based on the first input of the filter object and the first state of the filter object,
process the state equation to determine a second state of the filter object based on the first input of the filter object and the first state of the filter object,
retain the second state of the filter object in the dynamically typed text-based programming environment, and
invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained.

1. Summary

As will be discussed in more detail below, Claims 1, 2 and 5 are not rendered obvious by Denk in view of DFS because these references fail to disclose or suggest "**invoking a first call to a filtering method of a filter object to... retain the second state of the filter object in the dynamically typed text-based programming environment**," and also fail to disclose or suggest "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained**," as recited in amended claim 1, upon which claims 2 and 5 depend.

These claims features recite that the state of the filter object can be retained between calls to the filtering method of the filtering object in a dynamically typed text-based programming environment. Such state retention is not known for filter objects in a dynamically typed programming environment. Conventional systems do not retain state for filter objects in a dynamically typed environment.

The claimed invention retains state for a filter object between calls in a dynamically typed environment. This enables input data to be split into pieces and such pieces processed in separate calls to the filtering method of the filter object.

2. Denk

Denk is concerned with errors that may arise in digital signal processing (DSP) algorithms in VLSI chips. In particular, Denk is concerned with rounding errors that may occur when fixed point computations are utilized in DSP components, such as digital filters (see [0005] and [0006]) "Signal processing typically involves numerous, sequential, iterative and recursive computations, during which uncompensated precision reduction errors accumulate, thereby degrading filter performance, possibly to the point of filter instability." (Lines 5-9 of [0006])

Denk summarizes the invention disclosed therein as follows at [0029]:

> The present invention includes methods and apparatus that reduce the precision of an input signal value having a first precision to an output signal having a second, lesser precision in a manner that greatly reduces, or substantially cancels, a precision reduction error signal typically inherent in prior art rounding techniques. By combining the input signal with a selectable bias, responsive to a preselected threshold rounding state, the rounding methods and apparatus according to the present invention provide an output signal that is substantially free of precision reduction error bias. In addition, where it is desired to produce a preselected signal offset, values for the selectable bias can be assigned to generate the offset. Such a signal offset may be useful to compensate for a undesirable pre-existing input signal bias, including correcting for precision reduction error biases injected during previous precision reduction operations. Also, it may be useful to impart an offset to an output signal, for example, to pre-condition the output signal for an anticipated bias arising from subsequent signal processing or in a communication channel.

Figure 6 of Denk depicts a method for converting a real-valued input signal X to an integer-valued reduced precision output value X. The method includes step 630 of performing selectable bias rounding. Denk notes that MATLAB® can be used to model the selectable bias rounding step and gives some example MATLAB® code for performing the step.

3. DFS

DFS is a group of webpages that were available on a website in 2002. The webpages concern some software designated as "Digital Filter Solutions" that was sold by Nuhertz Technologies, LLC. The Digital Filter Solutions software facilitated digital filter synthesis and analysis. The DFS webpages primarily discuss Z-transforms. A Z-transform converts a discrete time domain signal into a complex frequency domain representation. This transformation can be useful to implement a digital filter in the frequency domain. The software allowed a user first to develop a Z-transform and then generate code written in the C programming for implementing a digital filter for the Z-transform.

4. Unobviousness of claims 1, 2 and 5

Denk and DFS, alone or in any reasonable combination, fail to disclose or suggest "**invoking a first call to a filtering method of a filter object to…retain the second state of**

the filter object in the dynamically typed text-based programming environment," as recited in claims 1, upon which claims 2 and 5 depend. Denk is entirely silent about retaining the second state of the filter object after a call to a filtering method of a filter object. Denk mentions in passing that MATLAB®, a dynamically typed language, can be used to perform the selectable bias rounding step, but makes no mention of filter objects, retaining the state of filter object or calls to a filtering method of a filter object.

DFS also does not disclose or suggest a filter object in a dynamically typed text-based programming environment and does not suggest retaining state of the filter object after a call to the filtering method of the filter object. DFS discloses that software for implementing a digital filter can be generated from a Z transform. The software that is generated is in the C programming language, which is statically typed. DFS does not suggest retaining of the second state of the filter object in a dynamically type text-based programming environment.

Denk and DFS, alone or in any reasonable combination, also fail to disclose or suggest "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained**," as recited in claim 1. Neither of the references, alone or in combination, disclose or suggest invoking a second call to the filtering method using the second state that was retained after the first call to the filter method. Moreover, neither of these references disclose retaining the state for a filter object in a dynamically typed text-based programming environment. Therefore, Denk and DFS fail to disclose or suggest this claim feature.

Moreover, there does not appear to be a likelihood of success in combining Denk with DFS even if a person skilled in the relevant arts were to somehow combine Denk with DFS. Denk concerns hardware devices to be implemented on chips. It is not clear how the C code of DFS can be generated from such devices. Moreover, DFS appears to disclose a self-contained software package where the user must input data to generate the C code for a digital filter. It does not have a disclosure regarding how to generate C code for hardware devices.

Accordingly, the Applicants respectfully urge reconsideration of the rejection of claims 1, 2 and 5.

C. Claims 21-23, 26 and 27

Amended claim 21 reads as follows:

21. A computer-implemented method for generating an output of a system in response to an input to the system, the method comprising:

implementing the system using a dynamically typed text-based programming environment, the system including a filter object with a filtering method; and

using the system in the dynamically typed text-based programming environment, the system:

invoking a first call to the filtering method of the filter object to:

determine a first output of the system based on a first input to the system and a first state of the system;

determine a second state of the system based on the first input to the system and the first state of the system;

retain the second state of the system in the dynamically typed text-based programming environment so that the second state is available after the output of the system is determined; and

invoking a second call to the filtering method of the filter object to determine a second output of the system based on the second input to the system and the second state of the system that has been retained.

Claims 22, 23, 26 and 27 depend upon base claim 21 and thus, incorporate the claim features of claim 21.

Claims 21-23, 26 and 27 are not rendered obvious by Denk in view of DFS because these references alone or in any reasonable combination fail to disclose or suggest "**invoking a first call to the filtering method of the filter object to … retain the second state of the system in the dynamically typed text-based programming environment so that the second state is available after the output of the system is determined**," and "**invoking a second call to the filtering method of the filter object to determine a second output of the system based on the second input to the system and the second state of the system that has been retained**," as recited by claim 21.

As was discussed above relative to claims 1, 2 and 5, Denk and DFS, alone or in any reasonable combination, fail to retain the state of the system in the dynamically typed text-based

programming environment. In addition, Denk and DFS fail to disclose or suggest determining the second output of the system when a second call to the filtering method of the filter object is invoked based on the second state of the system that has been retained.

Accordingly, the Applicants respectfully request reconsideration of the claims 21-23, 26 and 27.

D. Claims 28-33 and 36

Amended independent Claim 28 reads as follows:

28. A computer readable medium holding instructions executable in a computer that provides a dynamically typed text-based programming environment, the instructions for:

providing an object, the object being an instance of a class;

invoking at least one method of the object to:

perform a first operation to determine a first output of the object based on a first input to the object and a first state of the object;

perform a second operation to determine a second state of the object based on the first input to the object and the first state of the object;

after completion of the first and second operations, **retain the second state of the object in the dynamically typed text-based programming environment**;

make the second state available after determining the output of the object; and

invoking the at least one method of the object to determine a second output of the object based on a second input to the object and the second state of the object that has been retained.

Claims 29-33 and 36 depend upon base claim 28.

Claims 28-33 and 36 are not rendered obvious by Denk in view of DFS because these references fail to disclose or suggest "**invoking at least one method of the object to…retain the second state of the object in the dynamically typed text-based programming environment**" and "**invoking the at least one method of the object to determine a second output of the object based on a second input to the object and the second state of the object that has been retained**," as recited in claim 28, upon which claims 29-33 and 36 depend. As has been discussed above, Denk and DFS, alone or in any reasonable combination, fail to disclose or suggest retaining the state of the object in the dynamically typed text-based programming environment between calls to a method of the object and they fail to disclose or

suggest invoking the at least one method of the object, to determine the second output using the second state of the object that has been retained, as recited in claim 28-33 and 36.

As such, the Applicants respectfully request reconsideration of the claims 28-33 and 36.

E. Claims 37, 39 and 44-47

Independent Claim 37 reads as follows:

37. A system for implementing a filter object, the system comprising:
a processor configured to process:
an output equation of the filter object processed in a first operation to determine a first output of the filter object based on a first input to the filter object and a first state of the filter object, and processed in a second operation to determine a second output of the filter object based on a second input to the filter object and a second state of the filter object;
a state equation of the filter object processed in a third operation to determine the second state of the filter object based on the first input to the filter object and the first state of the filter object; and
a memory for retaining the second state of the filter object in a dynamically typed text-based programming environment after completion of the first and third operations so that the second state is available after the output equation and the state equation are processed.

Claims 39, and 44-47 depend on claim 37.

Claims 37, 39 and 44-47 are not rendered obvious by Denk in view of DFS because these references fail to disclose or suggest "**a memory for retaining the second state of the filter object in a dynamically typed text-based programming environment after completion of the first and third operations so that the second state is available after the output equation and the state equation are processed**." As has been discussed above, Denk and DFS, alone or in any reasonable combination, fail to disclose the retaining of the state of the filter in a dynamically typed text-based programming environment so that the state is available after the output equation and the state equation are processed. Thus, the claims 37, 39 and 44-47 are not rendered obvious by Denk and DFS.

The Applicants respectfully urge reconsideration of the claims 37, 39 and 44-47.

III. Rejection of claims 4, 24, 25, 34, 42 and 43 under 35 USC § 103(a) as being unpatentable over Denk in view of DFS and Gay

Claims 4, 24, 25, 34, 42 and 43 stand rejected under 35 USC § 103(a) as being rendered obvious by Denk in view of DFS in further view of Gay. The Applicants respectfully traverse this rejection.

A. Claim 4

Claim 4 is not rendered obvious by Denk in view of DFS in further view of Gay. Claim 4 depends upon claim 1 and thus incorporates all the claim features of claim 1. As was discussed above, Denk and DFS fail to disclose or suggest "**invoking a first call to a filtering method of the filter object to … retain the second state of the filter object in the first dynamically typed text-based programming environment**," and "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained**," as recited in amended claim 1. Gay also fails to disclose or suggest these claim features.

Gay concerns a fast recursive least squares (FRLS) adaptive filter which provides lower computational complexity and provides dynamically adjusted regularization in real time without requiring restarting and adaptive filter (see column 1, lines 49-53). The Examiner cites Gay for disclosing the resetting of the state of the filter object that is retained in the filter object. The Examiner points to language of lines 5-26 of column 3 of Gay.

Gay is silent as to "**invoking a first call to a filtering method of the filter object to … retain the second state of the filter object in the first dynamically typed text-based programming environment**," and "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that**

has been retained," as recited in claim 1 upon which claim 4 depends. Gay does not address retaining the state of a filter object between calls to a filtering method of the filter object.

Accordingly, the Applicants respectfully urge reconsideration of the outstanding rejection of claim 4.

B. Claims 24 and 25

Claims 24 and 25 depends upon base claim 21 and thus incorporate all of the limitations of claim 21. As was discussed relative to claim 21, the combination of Denk in view of DFS and Gay fails to disclose or suggest "**invoking a first call to the filtering method of the filter object to... retain the second state of the system in the dynamically typed text-based programming environment so that the second state is available after the output of the system is determined**," and "**invoking a second call to the filtering method of the filter object to determine a second output of the system based on the second input to the system and the second state of the system that has been retained**," as recited in claim 21, upon which claims 24 and 25 depend. Gay also fails to disclose or suggest these claim features.

Hence, the Applicants respectfully urge reconsideration of the outstanding rejection of claim 24.

C. Claim 34

Claim 34 depends upon claim 28, which contains limitations similar to those discussed above for claims 1 and 21. Specifically, claim 28 recites "**invoking at least one method of the object to... retain the second state of the object in the dynamically typed text-based programming environment**," and "**invoking the at least one method of the object to determine a second output of the object based on a second input to the object and the second state of the object that has been retained**." Denk, DFS, and Gay, alone or in any reasonable combination, fail to disclose or suggest these claim features.

As such, the Applicants respectfully urge reconsideration of the outstanding rejection of claims 34.

D. Claims 42 and 43

Claim 42 and 43 depend upon base claim 37. Claim 37 recites "**a memory for retaining the second state of the filter object in a dynamically typed text-based programming environment after completion of the first and third operations so that the second state is available after the output equation and the state equation are processed**." Denk, DFS, and Gay, alone or in any reasonable combination, fail to disclose or suggest this limitation.

Accordingly, the Applicants respectfully urge reconsideration of the outstanding rejection of claims 42 and 43.

IV. Rejection of claims 8-13 and 15-20 under 35 §USC 103(a) as being unpatentable over Denk in view of DFS in further view of AutoCode Solutions

Claims 8-13 and 15-20 stand rejected under 35 under 35 §USC 103(a) as being rendered obvious by Denk in view of DFS and in further view of AutoCode Solutions. The Applicants respectfully traverse this rejection.

Claims 8-13 and 15-20 depend upon claim 1. As such, they incorporate all the claim features of claim 1.

As has been discussed above, Denk and DFS, alone or in any reasonable combination, fail to disclose or suggest "**invoking a call to a filtering method of the filter object to … retain the second state of the filter object in the first dynamically typed text-based programming environment**," and "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained**," as recited in claim 1.

AutoCode also fails to disclose or suggest these claim features. AutoCode Solutions is just additional webpages from the same website as DFS. The AutoCode Solutions pages note that C code may be generated for a digital filter based upon a Z transform. AutoCode Solutions

is entirely silent as to the claim features that have been quoted above. AutoCode Solutions discloses an implementation of a digital filter in a statically typed programming language, not a dynamically typed programming environment. Moreover, AutoCode Solutions does not discuss retaining the state of a filtering object between calls to a filter method of the filter object, as required by the quoted claim features.

Therefore, the Applicants respectfully urge reconsideration of the outstanding rejection of claims 8-13 and 15-20.

V. Rejection of Claim 14 under 35 §USC 103(a) as being unpatentable over Denk in view of DFS, AutoCode Solutions and in further view of Velazquez

Claim 14 stands rejected under 35 §USC 103(a) as being rendered obvious by Denk, DFS, AutoCode Solutions and Velazquez. The Applicants respectfully traverse this rejection.

Claim 14 depends upon base claim 1 and thus incorporates all of the claim features of claim 1. As has been discussed above, Denk, DFS and AutoCode Solutions, alone or in any reasonable combination, fail to disclose or suggest the claim features of "**invoking a call to a filtering method of the filter object to … retain the second state of the filter object in the first dynamically typed text-based programming environment**," and "**invoking a second call to the filtering method of the filter object to process the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object that has been retained**," as recited in claim 1.

Velazquez also fails to teach or suggest these claim features. Velazquez deals with a linearity error compensator that compensates for linearity errors, such as harmonic distortion and inter-modulation distortion, in electronic devices. The Examiner points to language noting that exponential coefficients can be implemented in firmware or software. This language does not disclose or suggest the quoted claim features.

Accordingly, the Applicants respectfully urge reconsideration of the outstanding rejection of claim 14.

VI. Rejection of Claim 35 under 35 §USC 103(a) as being unpatentable over Denk in view of DFS and in further view of Pickerd

Claim 35 stands rejected under 35 §USC 103(a) as being rendered obvious by Denk in view of DFS and in further view of Pickerd. The Applicants respectfully traverse this rejection.

Claim 35 depends upon independent claim 28 and thus incorporates the claim features of claim 28. Claim 28 includes the features of "**invoking at least one method of the object to… retain the second state of the object in the dynamically typed text-based programming environment**," and "**invoking the at least one method of the object to determine a second output of the object based on a second input to the object and the second state of the object that has been retained**."

Denk, DFS and Pickerd, alone and in any reasonable combination, fail to disclose or suggest these claim features. The failure of Denk and DFS to disclose these claim features has been discussed above relative to claim 28. Pickerd also fails to disclose these claim features. Pickered is concerned with a streaming distributed oscilloscope but does not disclose the quoted claim features.

As such, the Applicants respectfully urge reconsideration of the outstanding rejection of claim 35.

VII. Conclusion

Applicants have made a good faith effort to place the claims in a state proper for allowance. Applicants urge the Examiner to reconsider the outstanding rejection in view of the amendments and comments contained herein and pass the claims to allowance. The Examiner is urged to contact applicants' representative if this would help expedite the prosecution of this application.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-030RCE3. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. § 1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: April 27, 2010

Respectfully submitted,

Electronic signature: /Kevin J. Canning/
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant